

# Animaciones : Galería animada al cargar la página

**Autor:** Marina Vega Vázquez **Fecha:** 25/01/2023

- Animaciones : Galería animada al cargar la página
  - Estilos
    - Para el cuerpo y main
    - Para la galería - ul
    - Para las imágenes
  - Animación
  - Media Queries

En esta práctica emplearemos una animación con fotograma clave (keyframe) que se realizará al cargar la página

También emplearemos una hoja de estilos (OpenPROPS) que tiene definido una serie de variables predefinidas. Puedes ver información sobre ella en [OpenPROPS - https://open-props.style/](https://open-props.style/). Importamos el css poniendo:

```
@import "https://unpkg.com/open-props";
```

en nuestra hoja de estilos.

Partirás del siguiente código html:

```
<body>
  <header>
    <h1>Galería animada al cargar la página</h1>
  </header>
  <main>
    <ul>
      <li style="--x1: 2; --x2: 6; --y1: 1; --y2: 4;">
        
      </li>
      <li style="--x1: 6; --x2: 8; --y1: 2; --y2: 4;">
        
      </li>
      <li style="--x1: 1; --x2: 4; --y1: 4; --y2: 7;">
        
      </li>
      <li style="--x1: 4; --x2: 7; --y1: 4; --y2: 7;">
        
      </li>
      <li style="--x1: 7; --x2: 9; --y1: 4; --y2: 6;">
        
      </li>
      <li style="--x1: 2; --x2: 4; --y1: 7; --y2: 9;">
        
      </li>
    </ul>
  </main>
</body>
```

```

    </li>
    <li style="--x1: 4; --x2: 7; --y1: 7; --y2: 10;">
      
    </li>
    <li style="--x1: 7; --x2: 10; --y1: 6; --y2: 9;">
      
    </li>
  </ul>
</main>
</body>

```

Vamos a examinar previamente el código anterior. Como puedes ver en cada li estamos definiendo unas variables, estas variables determinan la posición de la imagen en el Grid indicando cada una el valor inicial y valor final del eje X o Y de la grilla. Ejemplo para

- indica que el li (y por tanto la imagen que la contiene) empieza en la columna 2 y termina en la 4 (grid-column: 2 / 6) y que empieza en la fila 1 hasta la 3 (grid-row: 1 / 4). Esto se realiza con el siguiente código css:

```

li {
  grid-column: var(--x1, auto) / var(--x2, auto);
  grid-row: var(--y1, auto) / var(--y2, auto);
}

```

Al usar var() con dos valores estamos diciendo que si la variable (ejemplo --x1) no existe se use como valor auto.

## Estilos

Definimos las siguientes variables:

```

--big-tile-size: 20vmin;
--scale: 0.4;
--rotation: 270deg;
--tile-size: calc(var(--big-tile-size) / 3);

```

Para el cuerpo y main

### Para el cuerpo

- **Color de fondo** variable --sand-1 de OpenPROPS
- **Ancho y alto** 100% del viewport
- **Posicionado** flexbox.

### Para el main

- **Ancho y alto** 80% y 60% respectivamente.

## Para la galería - ul

ul { list-style-type: none; display: grid; gap: 1vmin; grid-template: repeat(9, var(--tile-size)) / repeat(9, var(--tile-size)); position: fixed; top: 50%; left: 50%; transform: translate(-50%, -50%) scale(var(--scale)); }

- **Posicionamiento**
  - Grid. 9x9. Donde el tamaño es el indicado por la variable --tile-size.
  - Fijo. arriba 50% izquierda 50%.
- **gap** 1vmin. ¿Que es vmin?.
- **Transformación** movido -50% tanto del eje de las X como Y. Escalado --scale.

## Para los li

- **max-inline-size: 100%** La propiedad de CSS max-inline-size define el tamaño máximo horizontal o vertical de un elemento bloque dependiendo del modo de escritura.
- Boder redondeado empleando la variable --radius-2 de OpenPROPS.
- Color de fondo hsl(0 0% 75% / 0.2);
- Posicionado de forma relativa.
- Sin relleno.

## Para las imágenes

- Definimos, de nuevo, a nivel de la imagen las variables --rotation (-270 grados). Al hacer esto la animación a nivel de imagen se realizará con estos valores no con los valores definidos a nivel del root.
- **Alto y "Ancho mínimo"** 200%.
- **Ajuste de la imagen a la cuadrícula** aspect-ratio: 1. La propiedad CSS aspect-ratio establece una relación de aspecto preferida para la caja, que se utilizará en el cálculo de los tamaños automáticos y algunas otras funciones de diseño. La relación de aspecto es la proporción que tiene una imagen entre su ancho y su alto, y normalmente se expresa como X:Y, siendo X el ancho e Y el alto. La relación de aspecto más básica es 1:1, que es un cuadrado, ya que su ancho es igual a su alto.
- **Posicionamiento** Absoluto. Arriba 50% izquierda 50%.
- **Transformación** movido -50% tanto del eje de las X como Y.

## Animación

Definimos la animación para el ul y img.

En este caso creamos una animación con las siguientes características:

Nombre	Duración	animation-fill-mode	Tipo de desplazamiento
scale-up	1s	both	0% - 100%

**Animation-fill** Estilo para el elemento de destino al terminar la animación. **both** La animación seguirá las reglas tanto para adelante como para atrás, extendiendo las propiedades de animación en ambas direcciones.

**0%-100%** Es equivalente a usar from y to. Valor de inicio y valor final.

## Para el inicio.

Traladamos eje X	Traladamos eje Y	Escalado	Rotación
------------------	------------------	----------	----------

Traladamos eje X	Traladamos eje Y	Escalado	Rotación
-50%	-50%	--scale	0°

Para el fin

Traladamos eje X	Traladamos eje Y	Escalado	Rotación
-50%	-50%	1	--rotation

## Media Queries

Diseñamos la página para cualquier dispositivo tablet o móvil y escritorio hasta 1600px, de forma que pondremos la mitad del tamaño de cada carta. Por encima de 1600px usar el tamaño real de la imagen.